

# What is MLflow?

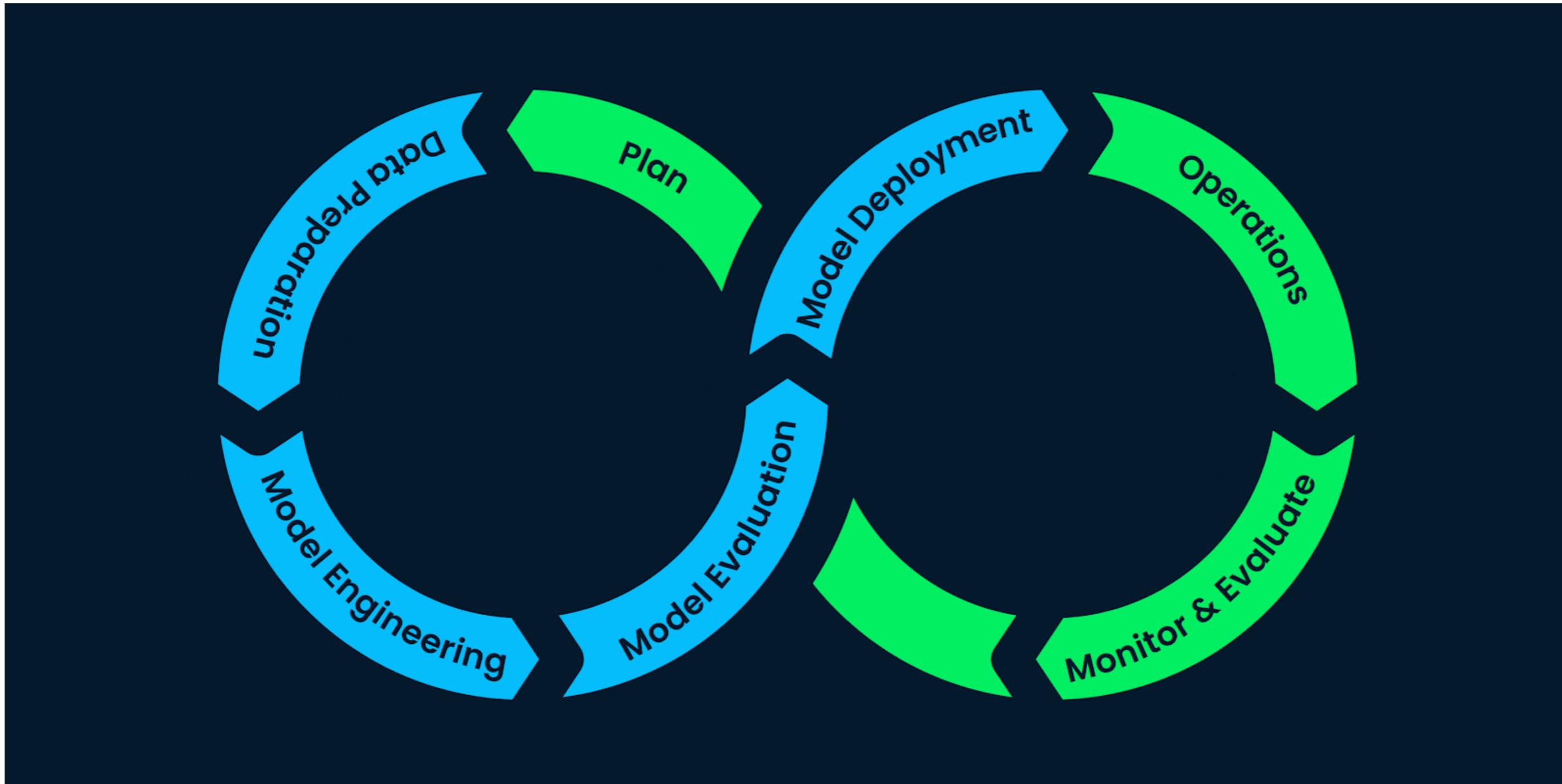
INTRODUCTION TO MLFLOW



**Weston Bassler**

Senior MLOps Engineer

# The machine learning lifecycle



<sup>1</sup> [datacamp.com](https://datacamp.com)

# Difficulties of machine learning

- **Tracking:**
  - Models and model metrics
- **Reproducibility:**
  - Platforms, environments and workspaces
- **Deployment:**
  - So many options and no standards

# What is MLflow?



"An open source platform for the machine learning lifecycle" - [MLflow.org](https://MLflow.org)

<sup>1</sup> [www.mlflow.org](https://www.mlflow.org)

# Components of MLflow

- **MLflow Tracking:**
  - Record metrics and parameters from training runs
  - Query data from experiments
  - Store models, artifacts and code
- **Model Registry:**
  - Store and version ML models
  - Load and deploy ML models
- **MLflow Models:**
  - Standardize models for deployment
  - Build customized models
- **MLflow Projects:**
  - Package ML code for reproducibility
  - Package ML code for repeatability

## Integrations with:



PyTorch

Keras



MONAI<sup>†</sup>

RAPIDS



python™



mleap

ONNX



XGBoost

LightGBM

spaCy

fast.ai



CatBoost

PYCARET

ALGORITHMIKA



OPTUNA

RAY

CONDA

kubernetes

docker



Amazon SageMaker

Azure Machine Learning

Google Cloud

databricks

<sup>1</sup> [www.mlflow.org](http://www.mlflow.org)

# MLflow experiments

## Experiments

+   Default Share

Search Experiments

	Experiment Name	Action
<input checked="" type="checkbox"/>	Default	
<input type="checkbox"/>	Scores Experiment	
<input type="checkbox"/>	Scores	
<input type="checkbox"/>	Unicorn Experiment	
<input type="checkbox"/>	Unicorn	
<input type="checkbox"/>	Unicorn Model	
<input type="checkbox"/>	5	
<input type="checkbox"/>	Test	
<input type="checkbox"/>	7	
<input type="checkbox"/>	Test 2	
<input type="checkbox"/>	9	
<input type="checkbox"/>	Test 3	
<input type="checkbox"/>	11	

## Default

Track machine learning training runs in experiments. [Learn more](#) X

Experiment ID: 0 Artifact Location: ./mlruns/0

> Description [Edit](#)

metrics.rmse < 1 and params.model = "tree" i

Sort: Created ▼ Columns ▼

Time created: All time ▼ State: Active ▼

Showing 3 matching runs

	Run Name	Created	Duration	metric_1	metric_2
<input type="checkbox"/>	rumbling-deer-742	3 months ago	2.0s	0.872	1.824
<input type="checkbox"/>	receptive-kit-255	3 months ago	2.0s	0.86	1.356
<input type="checkbox"/>	bright-gnu-469	3 months ago	2.0s	0.242	1.263

# Working with experiments

## MLflow Client

- Create Experiments

```
client.create_experiment("Name")
```

- Tag Experiments

```
client.set_experiment_tag("Name",  
    k, v)
```

- Delete Experiments

```
client.delete_experiment("Name")
```

## MLflow module

- Create Experiments

```
mlflow.create_experiment("Name")
```

- Tag Experiments

```
mlflow.set_experiment_tag(k, v)
```

- Delete Experiments

```
mlflow.delete_experiment("Name")
```

- Set Experiment

```
mlflow.set_experiment("Name")
```

# Starting a new experiment

```
import mlflow  
# Create new Experiment  
mlflow.create_experiment("My Experiment")  
# Tag new experiment  
mlflow.set_experiment_tag("scikit-learn", "lr")  
# Set the experiment  
mlflow.set_experiment("My Experiment")
```

# **Let's practice!**

**INTRODUCTION TO MLFLOW**

# MLflow Tracking

INTRODUCTION TO MLFLOW



**Weston Bassler**

Senior MLOps Engineer

# Tracking data about models



<sup>1</sup> istock.com

# What is MLflow Tracking?

- Model Metrics
  - F1, Recall, Accuracy, MSE, etc...
- Parameters
  - library specific
- code
  - `train.py`
- other artifacts
  - tokenizers, pickle, etc...

# Training runs

- How MLflow is organized
- New run equals new model training
- A run is placed within an experiment
- Invoked via `mlflow.start_run()`



<sup>1</sup> unsplash.com

# Starting a training run

```
import mlflow  
  
# Start a run  
mlflow.start_run()
```

```
<ActiveRun: >
```

```
# End a run  
mlflow.end_run()
```

# Setting a training run variable

```
import mlflow  
# Set experiment  
mlflow.set_experiment("My Experiment")  
# Start a run  
run = mlflow.start_run()  
# Print run info  
run.info
```

```
<RunInfo: artifact_uri='./mlruns/0/9de5df4d19994546b03dce09aefb58af/artifacts',  
end_time=None, experiment_id='31', lifecycle_stage='active',  
run_id='9de5df4d19994546b03dce09aefb58af', run_name='big-owl-145',  
run_uuid='9de5df4d19994546b03dce09aefb58af', start_time=1676838126924,  
status='RUNNING', user_id='user'>
```

# Logging to MLflow Tracking

- **Metrics**

- `log_metric("accuracy", 0.90)`

- `log_metrics({"accuracy": 0.90, "loss": 0.50})`

- **Parameters**

- `log_param("n_jobs", 1)`

- `log_params({"n_jobs": 1, "fit_intercept": False})`

- **Artifacts**

- `log_artifact("file.py")`

- `log_artifacts("./directory/")`

# Logging a run

```
import mlflow  
# Set Experiment  
mlflow.set_experiment("LR Experiment")  
  
# Start a run  
mlflow.start_run()  
  
# Model Training Code here  
lr = LogisticRegression(n_jobs=1)  
  
# Model evaluation Code here  
lr.fit(X, y)  
score = lr.score(X, y)
```

```
# Log a metric  
mlflow.log_metric("score", score)  
  
# Log a parameter  
mlflow.log_param("n_jobs", 1)  
  
# Log an artifact  
mlflow.log_artifact("train_code.py")
```

# Open MLflow UI

```
# Open MLflow Tracking UI  
mlflow ui
```

Go to: <http://localhost:5000>



# Tracking UI experiment view

## LR Experiment

[Share](#)

Experiment ID: 37    Artifact Location: ./mlruns/37

› Description [Edit](#)

<input type="text"/> Q metrics.rmse < 1 and params.model = "tree" <span style="margin-left: 20px;">(i)</span> <span style="margin-left: 20px;">Sort: Created</span>				⋮ <span style="margin-left: 20px;">Refresh</span>	
<input type="button" value="Columns"/>					
<input type="button" value="Time created: All time"/>		<input type="button" value="State: Active"/>		Showing 1 matching run	
Run Name	Created	Models	Metrics	Parameters	
<input type="checkbox"/>	silent-slug-662	1 minute ago	-	score	n_jobs
<hr/>					
<hr/>					

# Tracking UI run view

[LR Experiment >](#)

## silent-slug-662

Run ID: a410480d4ccc4601904085b5651483b4

Date: 2023-02-20 08:14:05

Source: 

User: [weston](#)

Duration: 1.5min

Status: FINISHED

Lifecycle Stage: [active](#)

› Description [Edit](#)

▼ Parameters (1)

Name	Value
n_jobs	1

▼ Metrics (1)

Name	Value
score 	0.951

› Tags

▼ Artifacts

 train\_code.py

# **Let's practice!**

**INTRODUCTION TO MLFLOW**

# Querying runs

INTRODUCTION TO MLFLOW



**Weston Bassler**

Senior MLOps Engineer

# Model data

Model data is the data used to train, validate, and test machine learning models. It typically includes training data, validation data, and test data. Training data is used to teach the model how to make predictions. Validation data is used to tune the model's hyperparameters and prevent overfitting. Test data is used to evaluate the model's performance on unseen data.

# Runs data

## Insurance Experiment

Experiment ID: 27    Artifact Location: ./mlruns/27

› Description [Edit](#)

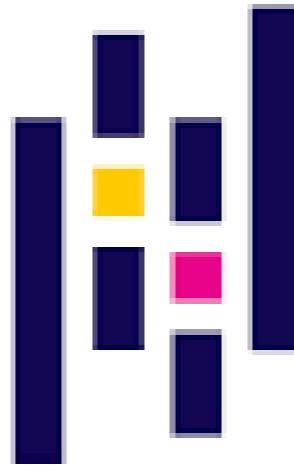
Q metrics.rmse < 1 and params.model = "tree"   Sort: accuracy\_score 

Time created: All time  State: Active 

Metrics									
	Run Name	Created	accuracy_score	example_count	f1_score	false_negatives	false_positives	precision_score	recall_score
<input type="checkbox"/>	wise-mole-318	 shap, 1 mo	0.621	335	0.623	65	62	0.629	0.618
<input type="checkbox"/>	invincible-lark-929	 shap, 1 mo	0.621	335	0.623	65	62	0.629	0.618
<input type="checkbox"/>	sedate-fawn-631	 shap, 1 mo	0.621	335	0.623	65	62	0.629	0.618
<input type="checkbox"/>	awesome-moth-445	 shap, 1 mo	0.621	335	0.623	65	62	0.629	0.618
<input type="checkbox"/>	fun-grub-597	 shap, 1 mo	0.621	335	0.623	65	62	0.629	0.618
<input type="checkbox"/>	...	...	...	...	...	...	...	...	...

# Searching runs

```
mlflow.search_runs()
```



pandas

<sup>1</sup> <https://pandas.pydata.org/>

# Output format

#	Column	Non-Null Count	Dtype
0	run_id	6 non-null	object
1	experiment_id	6 non-null	object
2	status	6 non-null	object
3	artifact_uri	6 non-null	object
4	start_time	6 non-null	datetime64[ns, UTC]
5	end_time	5 non-null	datetime64[ns, UTC]
6	metrics.test	1 non-null	float64
7	metrics.metric_2	3 non-null	float64
8	metrics.metric_1	3 non-null	float64
9	params.param_1	3 non-null	object
10	params.random_state	3 non-null	object
11	params.n_estimators	3 non-null	object
12	tags.mlflow.user	6 non-null	object
13	tags.mlflow.runName	6 non-null	object
14	tags.mlflow.source.type	6 non-null	object
15	tags.mlflow.source.name	6 non-null	object

# Filtering run searches

- `max_results` - maximum number of results to return.
- `order_by` - column(s) to sort in `ASC` ending or `DESC` ending order.
- `filter_string` - string based query.
- `experiment_names` - name(s) of experiments to query.

# Tracking UI

## Insurance Experiment

Experiment ID: 27    Artifact Location: ./mlruns/27

> Description [Edit](#)

Q metrics.rmse < 1 and params.model = "tree"   Sort: accuracy\_score 

Time created: All time  State: Active  S

Metrics								
	Run Name	Created	Duration	accuracy_score	f1_score	false_negative	false_positives	precision_score
<input type="checkbox"/>	wise-mole-318	 3 months ago	10.4s	0.621	0.623	65	62	0.629
<input type="checkbox"/>	powerful-shoat-853	 3 months ago	5.0s	0.621	0.623	65	62	0.629
<input type="checkbox"/>	amazing-penguin-22	 3 months ago	4.6s	0.537	0.485	97	58	0.557
<input type="checkbox"/>	traveling-snipe-808	 3 months ago	5.6s	0.537	0.485	97	58	0.557

# Search runs example

```
import mlflow
# Filter string
f1_score_filter = "metrics.f1_score > 0.60"
# Search runs
mlflow.search_runs(experiment_names=["Insurance Experiment"],
                    filter_string=f1_score_filter,
                    order_by=["metrics.precision_score DESC"])
```

# Example output

```
# Search runs from Insurance Experiment  
mlflow.search_runs(experiment_names=["Insurance Experiment"],  
                    filter_string=f1_score_filter,  
                    order_by=["metrics.precision_score DESC"])
```

```
run_id experiment_id ... tags.mlflow.source.type tags.mlflow.user  
0 90407e29a5aa4a31954bed874c7d4337 27 ... LOCAL user  
1 c335c0b16a5d4cf398aaa7189362b577 27 ... LOCAL user
```

# **Let's practice!**

**INTRODUCTION TO MLFLOW**